DETAILED DESCRIPTION

FIG. 2a illustrates, in block diagram form, a local switching network 200 employing the present invention. Local switch network 200 includes a switching fabric 202 (e.g., a cross-bar switching fabric) coupled to line cards 204 through 208. Line card 204 is coupled to end devices 210 through 214, line card 206 is coupled to end devices 216 through 220, and line card 208 is coupled to end devices 224 through 226.

The local switching network 200 shown in FIG. 2a may employ one of many different communication protocols enabling data communication between one or more end devices 210 through 226 via line cards 204 through 208 and switching fabric 202. FIG. 2a will be described with reference to a communications protocol in which end devices communicate with each other by transferring variable sized data frames or packets via line cards and the switching fabric. It is noted end devices (e.g., end devices 210 and 212) can locally communicate with each other without having to transmit data packets via fabric 202. The present invention will be described with reference to end devices communicating with each other via fabric 202.

Each data packet in a data communication includes one or more lines of data. The lines of the data packet may be transmitted during consecutive data transmission cycles of the link between the line card and the switching fabric. Lines of the data packet may also be transmitted during non-consecutive transmission cycles of the link between the line card and the switching fabric.

Line cards 204 through 208 are coupled to switching fabric 202 via one or more data links. In FIG. 2a, line card 204 is coupled switching fabric 202 via downlink and uplink 228 and 230, respectively, line card 206 is coupled switching fabric 202 via downlink and uplink 232 and 234, respectively, and line card 208 is coupled switching fabric 202 via downlink and uplink 236 and 238, respectively. Data is transferred between line card 204 and end devices 210 through 214 via common bus 240, between line card 206 and end devices 216 through 220 via common bus 242, and between line card 208 and end devices 222 through 226 via common bus 244.

FIG. 2b illustrates in block diagram form one embodiment of line card 204. Line cards 206 and 208 may take similar form. In FIG. 2b, line card 204 includes a data buffer 250, a formatter 252, a control circuit 254, and a multiplexer 256. Multiplexer 256 selectively couples an output of formatter 252 and common bus 240 to uplink 230 in response to a control signal generated by control circuit 254. Formatter 252 generates control codes to be transmitted to switching fabric 202. Although not shown, a separate formatter may be present in line card 204. The separate formatter may be provided to modify the first line of one or more data packets of a stream received from common bus 240 before the data packets are transmitted to the fabric via multiplexer 256. Control circuit couples the output of formatter 252 to uplink 232 in response to formatter 252 generating a control code. Otherwise, control circuit 254 couples common bus 240 to uplink 230 thereby allowing data input to line card 204 to be transmitted to switching fabric 202. Common bus 240 may take form in a bi-directional bus. In this embodiment, a switch may be provided (not shown) for selectively coupling the data buffer 250 or the multiplexer 256 to the bi-directional bus. In the alternative, common bus 240 may take form in a pair of unidirectional buses respectively coupled to buffer 250 and multiplexer 256.

Data buffer 250 may take form in a FIFO buffer. Buffer 250 is configured to receive and store data transmitted from switching fabric 202 via downlink 228. Data is stored within data buffer 215 prior to output on to common bus 240 for subsequent delivery to one of the end devices 210 through 214. Although not shown, a second buffer may be present in line card 204. The second buffer may store data packets received from common bus 240 before they are transmitted to fabric 202 via multiplexer.

Data buffer, in one embodiment, may include circuitry for generating information relating to the quantity of data stored therein. This information may be generated in any one of a number of different methods. For example, circuitry in buffer 250 may keep a running total of the data lines input to buffer and a running total of the data lines output from buffer 250. During each cycle of a system clock (not shown in FIG. 2b) the circuitry may subtract the running total of data lines output from the running total of data lines input, the difference provided to formatter 252 as q(t), the quantity of data stored in buffer 250 at time t. Alternatively, buffer 250 may provide the running totals to formatter 252 for subsequent calculation of q(t).

Formatter 252 is configured to receive the information relating to the quantity of data stored within buffer 250. For purposes of explanation, formatter 252 will be described as receiving q(t). In one embodiment, formatter 252 receives q(t) at each cycle of the system clock. Formatter 252 is configured to generate a plurality of control codes for transmission to switching fabric 202 via multiplexer 256 and uplink 230. As will be more fully described below, one of the control codes, a variable data transmit rate control code (hereinafter referred to as variable-rate control code), variably controls the rate at which data is transmitted from switching fabric 202 for storage within data buffer 250. In contrast to the prior art described above, formatter 252, through generation and transmission of the variable-rate control code, may vary the rate at which data is transmitted to buffer 250. The rate may vary from a full data transmit rate to zero data transmit rate, and at rates therebetween. For example, formatter 252 may generate a variable-rate control code instructing switching fabric 202 to transmit data at 0%, 12.5%, 25%,....100% of the full rate. Formatter 252 generates the variable-rate control code in response to receiving q(t). Accordingly, line card 204 may signal switching fabric 202 the status of its data buffer 250 by adjusting the rate at which data is transmitted to buffer 250 for storage therein. In one embodiment, the variable-rate control codes are fixed. In another embodiment, the variable-rate control codes are programmable. In this latter embodiment, the variable-rate control codes may be set by software operating in, for example, formatter 252.

Formatter 252 monitors the quantity of data stored within buffer 250 using q(t) provided therefrom. Although not shown in FIG. 2b, circuitry for monitoring buffer 250 in formatter 252 may take form in hardware, software, or a combination of hardware and software. Formatter 252, in one embodiment, compares q(t) against several programmable thresholds Q(1) - Q(n) during each cycle of the system clock provided to line card 204. Thresholds Q(1) - Q(n) represent increasing quantities of data ranging from Q(1) equal to zero to Q(n) equal to a percentage (e.g., 98%) of the total amount of data capable of being stored in buffer 250. When q(t) increases beyond any of the thresholds Q(1) - Q(n), then at that time formatter 252 generates and sends a variable-rate control code to switching fabric 202 instructing it to slow the rate at which it transmits data for storage in buffer 250. Before the variable-rate control code is sent, control 254 instructs multiplexer 256 to couple formatter 252 to uplink 230. Once the the variable-rate control code is transmitted to fabric 202, control 254 instructs multiplexer to couple